1. Consider the following PRAM CRCW algorithm FACTORIAL\( (L,n) \) that stores the values \( n \ldots 1 \) in a singly-linked list \( L \) on \( n \) processors, where \( y(i) \) refers to the number stored in processor \( i \). The algorithm returns \( n! \) using the List-Prefix approach.

\[
\text{FACTORIAL}(L,n) \\
1 \quad \text{while next}(i) \neq \text{NIL} \text{ for some processor } i \\
2 \quad \text{foreach processor } i, \text{ in parallel} \\
3 \quad \quad \text{if next}(i) \neq \text{NIL} \\
4 \quad \quad \quad \text{then } \text{__________________________} \\
6 \quad \text{return } y(n) \\
\]

(a) (3 points) Show the two missing lines of pseudocode from the FACTORIAL algorithm above.

(b) (3 points) Given the initial linked-list below for computing 5!, show the linked-list after each iteration of the main loop of FACTORIAL.

```
5 <-> 4 <-> 3 <-> 2 <-> 1
```
(c) (1 point) Give the asymptotic running time of the parallel FACTORIAL algorithm.

(d) (1 point) Give the asymptotic running time of the best serial algorithm for computing $n!$.

(e) (1 point) Give the speedup of the parallel FACTORIAL algorithm.
2. Suppose you select the two prime numbers \( p = 11 \) and \( q = 13 \) to generate public and secret keys in the RSA cryptosystem.

(a) (1 point) What is the formula and value of \( n \)?

(b) (1 point) What is the formula and value of \( \phi(n) \)?

(c) (2 points) What is the value of the smallest odd \( e > 1 \) that is relatively prime to \( \phi(n) \) from part (b)? Show your work.

(d) (2 points) What is the value of \( d \) that is the multiplicative inverse of \( e \) from part (c), modulo \( \phi(n) \) from part (b)? Show your work.

(e) (2 points) Give the public and secret keys generated from the above scenario.
3. Consider the string matching problem of finding all occurrences of pattern \( P = abcab \) in the text \( T = aaabcabad \), where \( \Sigma = \{a, b, c, d\} \).

(a) (2 points) Give the prefix function \( \pi \) for the pattern \( P \).

(b) (2 points) Give the bad character function \( \lambda \) for the pattern \( P \) and alphabet \( \Sigma \).

(c) (2 points) Give the good suffix function \( \gamma \) for the pattern \( P \).

(d) (1 point) Using the Knuth-Morris-Pratt algorithm how far is the pattern shifted after the first character mismatch?

(e) (1 point) Using the Boyer-Moore algorithm how far is the pattern shifted after the first character mismatch?